

1. Pendahuluan

1.1 Latar Belakang

Pada *game* action, banyak terdapat interaksi antara karakter yang satu dengan yang lainnya. Jika antara karakter terjadi interaksi maka seharusnya terdapat respon yang cepat dan tepat dari masing-masing karakter. Adanya deteksi tubrukan (*collision detection*) menyebabkan interaksi antara karakter menjadi terlihat lebih nyata. Terjadinya *collision detection* pada *game* action seharusnya menggunakan rentang waktu yang cukup singkat dan tepat.

Terdapat beberapa metode yang bisa digunakan untuk mendeteksi terjadinya tubrukan, yaitu metode irisan antar objek, multi-pass atau hybrid, rectangle, circle, ellips, multibounding box, dan multiline. Metode irisan antar objek merupakan metode yang paling sering digunakan. Tetapi hasilnya kurang detail sehingga masih terdapat sisa jarak ketika karakter saling bertabrakan. Kemudian terdapat metode multi-pass atau hybrid, yaitu sebuah metode yang mengeliminasi bagian karakter yang memiliki jarak yang jauh dengan bagian karakter lainnya.[13]

Metode Rectangle merupakan sebuah metode *collision detection* di mana pada setiap karakternya terdapat batas luar berupa segi empat. Tabrakan antar karakter terjadi apabila batas luar yang terdapat pada masing-masing karakter saling bersinggungan. Kekurangan yang terdapat pada teknik ini adalah masih terdapat banyak sisa antara karakter dengan batas luar sehingga tubrukan yang terjadi tidak terlihat secara detail. Metode lainnya yang juga banyak dipakai adalah Circle. Metode ini menggunakan lingkaran sebagai batas luar dari sebuah karakter. Kekurangan pada metode ini adalah tidak cocok untuk digunakan pada karakter yang panjang dan tipis karena mudah terjadi kesalahan pada saat pendeteksian tubrukan dan juga terdapat sisa antara karakter dengan batas luar sehingga tubrukan yang terjadi tidak terlihat secara detail.

Jika pada *game* 3D, elips yang kemudian berubah menjadi elipsoid, banyak digunakan maka pada *game* 2D elips jarang digunakan karena banyak terdapat sisa ruang antara karakter dengan batas luar yang berbentuk elips. Sedangkan untuk metode multibounding box adalah metode di mana terdapat banyak bounding box dan berukuran lebih kecil serta disesuaikan dengan ukuran karakter. Kekurangan dari metode multibounding box adalah masih terdapat sisa ruang antara karakter dengan batas luar tersebut. Metode Multiline atau Polygon merupakan sebuah metode dengan membuat sebuah garis pada sebuah sisi dari karakter. Garis tersebut kemudian menjadi batas luar dari karakter tersebut. Proses tubrukan pada Multiline dihitung jika terdapat irisan antara garis pada karakter satu dengan karakter yang lainnya.[5,6,11]

Karena pada *game* fighting, dimana banyak terdapat interaksi antar karakter, dibutuhkan sebuah *collision detection* yang detail dan tepat serta memiliki waktu komputasi yang cepat maka digunakan metode multiline. Selain itu, metode multiline merupakan metode yang memiliki waktu komputasi yang kecil karena perhitungan interaksi antar karakter adalah dengan perhitungan garis singgung antara batas luar karakter tersebut.[2,3]

Oleh karena itu, pada tugas akhir ini penulis memilih menggunakan metode multiline dan mengangkat studi kasus *Game Fighting* untuk menjadi objek penelitian penulis.

1.2 Perumusan Masalah

Permasalahan yang menjadi objek penelitian penulis, yaitu:

1. Berapa besar jarak yang tersisa antara objek sebenarnya dengan polygon yang membungkusnya?
2. Berapa besar waktu komputasi jika line diperbanyak?
3. Berapa besar waktu komputasi jika semakin banyak objek yang ada?
4. Berapa besar *step* ketika batasan *frame per second* berada dibawah 24 *frame per second* dan berapa waktu komputasinya?

1.3 Tujuan

Tujuan dari tugas akhir ini adalah:

1. Membuktikan bahwa penambahan objek dan berkurangnya *step* akan membuat rata-rata FPS menurun serta penggunaan CPU meningkat.
2. Dari banyak multiline dengan *step* yang berbeda-beda bisa didapatkan multiline terbaik berdasarkan *coverage* terhadap piksel sebenarnya, rata-rata FPS dan rata-rata penggunaan CPU dalam keadaan jumlah objek paling sedikit hingga jumlah objek paling banyak.
3. Membuktikan apakah hipotesis yang menyatakan bahwa multiline dengan *step* yang semakin kecil adalah multiline yang terbaik.

1.4 Batasan Masalah

Yang menjadi batasan masalah dari tugas akhir ini adalah:

1. Waktu yang dihitung adalah waktu terjadinya *collision* dan pada saat rendering gambar karakter tersebut.
2. Penambahan objek atau *line* berhenti hingga waktu komputasi mencapai 1 detik.
3. Perhitungan waktu komputasi hanya pada satu computer dengan satu kartu grafik dan prosesor dengan satu inti.
4. Waktu komputasi yang dihitung akan direpresentasikan dalam bentuk persentase penggunaan CPU.

1.5 Metodologi Penyelesaian Masalah

Metodologi yang digunakan dalam menyelesaikan tugas akhir ini, yaitu:

1. Studi kepustakaan, yaitu dengan mempelajari literatur-literatur yang ada, yang berkaitan dengan permasalahan yang meliputi:
 1. Konsep *collision detection*.
 2. Konsep *collision detection* menggunakan *multiline*.
2. Perumusan masalah, yaitu dengan mendalami materi yang digunakan untuk mendefinisikan masalah, yaitu:
 1. Perhitungan jarak pada bagian depan yang langsung terkena *collision*.

2. Besar waktu komputasi jika objek ditambah atau line ditambah.
3. Besar penggunaan CPU tertinggi, serta FPS dibawah 24 terdapat pada penambahan objek atau line beberapa.
3. Analisa Kebutuhan Sistem dan Perancangan Perangkat Lunak, yaitu melakukan analisa terhadap model implementasi yang dibangun dengan tujuan memahami secara jelas proses yang dilakukan pada sistem tersebut, serta perancangan dengan menggunakan konsep analisis dan desain yang berorientasi objek.

Deskripsi Sistem



Pada sistem ini, pertama kali sistem menentukan titik-titik pada setiap karakter untuk dijadikan polygon yang kemudian akan disimpan di dalam memori. Setelah karakter memiliki titik-titik tersebut, permainan dimulai dan pada saat itu perhitungan waktu komputasi, persentase penggunaan CPU, serta FPS dimulai. Kemudian proses akan dilakukan berulang-ulang hingga besar jarak antar titik adalah 5.

4. Implementasi Perancangan Perangkat Lunak, yaitu implementasi secara *coding* berdasarkan analisis dan desain dengan menggunakan bahasa pemrograman Java dibantu dengan *framework* GTGE. Dengan spesifikasi komputer sebagai berikut:
 - a. RAM 2 GB
 - b. Hardisk 160 GB
 - c. Prosesor Intel Celeron M 1.8 GHz
 - d. Kartu Grafis Intel 965 GM Express Chipset Family
 - e. *Sound card* Realtek
 - f. Operating System Microsoft Windows XP MSDNAA
5. Uji coba dan analisa system, yang meliputi:
 1. Mengimplementasikan *multiline* pada setiap objek yang ada pada studi kasus tersebut.
 2. Menghitung seberapa besar jarak antara bentuk karakter yang sebenarnya.
 3. Menghitung seberapa besar waktu komputasi, persentase CPU, serta FPS jika *line* ditambahkan.
 4. Menghitung seberapa besar waktu komputasi, persentase CPU, serta FPS jika objek ditambahkan.
 5. Pada penambahan objek beberapa atau line beberapa sehingga *frame per second* dibawah 30 fps serta hitung besar waktu komputasi.
6. Penyusunan laporan tugas akhir dan kesimpulan akhir.

1.6 Sistematika Penulisan

Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

BAB I Pendahuluan

Bab ini berisi latar belakang, perumusan masalah, tujuan, hipotesis, batasan masalah, metodologi penyelesaian masalah, serta sistematika penulisan buku tugas akhir

BAB II Landasan Teori

Bab ini berisi uraian teori mengenai *Collision Detection*, Metode pada *Collision Detection*, Waktu Komputasi, dan *Frame Rate*.

BAB III Perancangan dan Implementasi

Bab ini berisi rancangan algoritma yang digunakan untuk membuat polygon, mendeteksi tubrukan, serta perhitungan waktu komputasi serta *frame rate*.

BAB IV Analisis Hasil Implementasi

Bab ini berisi uraian hasil implementasi serta menganalisis parameter tingkat komputasi tertinggi, terendah, serta parameter *frame rate*.

BAB V Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan yang didapat dari pelaksanaan tugas akhir ini serta saran-saranyang diperlukan untuk perbaikan maupun pengembangannya lebih lanjut.